基于差异性采样的流数据聚类算法 *

邱云飞，孙梦冉†

(辽宁工程技术大学 软件学院, 辽宁 葫芦岛 125105)

摘　要：针对传统聚类算法对流数据进行聚类时面临时间复杂度高，存储空间需求大以及准确度较低的问题，提出一种基于差异性采样的流数据聚类算法。首先利用差异性采样法对流数据进行采样并用样本点构造核矩阵，然后利用核模糊 C 均值聚类算法对核矩阵中的点进行聚类得到一个带有标记的样本核矩阵，最后利用带有标记的样本核矩阵对流数据中的点进行划分。同时利用衰退聚类机制，实时更新样本核矩阵。实验结果表明，相比于传统聚类算法，该算法实现了更低的时间复杂度，同时实时聚类，得到较为理想的聚类结果。

关键词：差异性采样；衰退聚类机制；核模糊 C 均值；流数据；时间复杂度

中图分类号：TP391.9　　doi: 10.3969/j.issn.1001-3695.2017.12.0808

Stream data clustering algorithm based on differential sampling

Qiu Yunfei, Sun Mengran†

(*College of Software Liaoning Technical University*, *HuLudao Liaoning* 125105, *China*)

Abstract: Concerning the problems of high time complexity, large storage space requirements and low accuracy when traditional clustering algorithm cluster stream data, this paper proposed a kind of stream data clustering algorithm based on differential sampling. First, it used the differential sampling method sampled stream data, and used sample points to construct kernel matrix. Then it used kernel fuzzy C-means clustering algorithm clustered the data points in the kernel matrix, obtained a marked sample kernel matrix. Finally, using the marked kernel matrix divided the stream data. Meanwhile, this paper adopted the fading cluster mechanism to update kernel matrix in real time. Experimental results show that compared with the traditional clustering algorithm, the proposed algorithm achieves lower time complexity, real-time clustering at the same time, get the ideal clustering result.

Key words: differential sampling; fading cluster mechanism; kernel fuzzy C-means; stream data; time complexity

0　引言

近年来，对流数据进行聚类分析成为数据挖掘领域中的热点问题。但由于流数据到达的实时性、数据结构的不稳定性、数据量的无限性，利用传统聚类算法很难进行有效的聚类。为了解决以上问题，Aggarwal 等人[1]提出一种流聚类算法（CluStream 算法）。该算法通过在线维护(2d+3)维的微簇产生高质量的簇提升流数据聚类的准确性及效率，但对高维数据处理效率不高。文献[2]提出一种自适应非线性流聚类算法，应用核异常检测方法按照时间的局部性将流数据分成若干部分，并对每一部分进行聚类，自适应选取具有代表性的部分作为初始的类对流数据中的其他点进行聚类。该算法虽然减小了时间复杂度及对存储空间的利用，但没有考虑数据点本身的数据信息在流数据中的影响程度，因此聚类效果并不理想。文献[3]提出一种基于密度与网格的流聚类算法，将数据映射到网格空间中对应的网格之中，根据密度在网格空间中进行聚类。该算法利用数据本身的特性对流数据进行聚类，但是由于流数据数量的无限性，导致该算法的时间复杂度较高。文献[4]提出一种基于采样的流聚类算法（approximate kernel fuzzy C-means,AKFCM），对流数据进行随机采样并聚类。该算法大大降低了时间复杂度，但准确率较低。与此同时，也出现了针对不同需求的流聚类算法[5,6]。例如文献[5]针对数据流中流速的变化，在基于在线和离线聚类框架的基础上提出了基于动态滑动窗口的流聚类算法 DSC，使得滑动窗口大小可随数据流流速动态改变，同时设定了窗口改变阈值避免窗口的频繁变化。该算法对流数据的处理效率较高，但未考虑到流数据本身数据信息对聚类结果的影响，因此聚类准确度并不高。文献[6]对经典流数据聚类算法 CluStream 与经典密度聚类算法 DBSCAN[7]进行总结与改进，提出了适用于入侵检测环境的流数据聚类算法。该算法能够实现对流数据的实时聚类，但不能准确反映新流入数据的特征，

收稿日期：2017-12-18；修回日期：2018-01-29　　基金项目：国家自然科学基金资助项目（61404069）；辽宁省教育厅科学研究项目（LJYL048）

作者简介：邱云飞（1976-），男，教授，博士，主要研究方向为数据挖掘、智能数据处理；孙梦冉（1992-），女（通信作者），硕士研究生，主要研究方向为数据挖掘、智能数据处理（2456876943@qq.com）．

因此聚类性能较低。

针对以上问题，本文提出一种基于差异性采样的流数据聚类算法。首先采用统计杠杆分数(statistical leverage scores)[8]对流数据中的点进行采样，其次用样本点构造核矩阵，然后应用核模糊C均值聚类算法对样本核矩阵中的点进行聚类得到一个带有类别标记的核矩阵，然后用带标记的样本核矩阵对流数据中的点进行实时划分，最后利用衰退聚类机制(fading cluster mechanism)[2]删除不再具有代表性的类别，实时更新数据模型。

1 基础理论

1.1 模糊C均值聚类算法

模糊聚类算法是一种根据隶属度值最大原则来划分类别的数学方法，每个样本点以不同隶属度值同时属于多个类，最终将该点聚到对应隶属度值最大的类中，使得被聚到同一类中的数据对象之间相似度最大，不同类数据对象之间相似度最小。模糊C均值聚类算法(fuzzy C-means,FCM)[9]步骤为：首先随机初始化每个数据与各个类的隶属度得到初始隶属度矩阵；然后根据隶属度计算每一个类的聚类中心，接着更新隶属度矩阵。如此迭代，直到各个类的聚类中心不再发生变化或者隶属度值变化的绝对值低于设定阈值，算法结束。模糊C均值聚类算法的目标函数为

$$J_{FCM}=\sum_{j=1}^{C}\sum_{i=1}^{N}u_{ij}^{m}\left\|x_i-v_j\right\|^2 \tag{1}$$

$$\sum_{j=1}^{C}u_{ij}=1,u_{ij}\in[0,1] \tag{2}$$

其中：给定数据集 $X=\{x_1,x_2,\cdots,x_i,\cdots,x_N\}$；$C$ 为聚类个数；N 为样本个数；$U=\left[u_{ij}\right]_{C\times N}$ 为隶属度矩阵；u_{ij} 为数据点；x_i 隶属于第 j 类的隶属度值；v_j 为第 j 类的聚类中心；m 为加权指数，也称平滑因子，控制模式在模糊类间的分享程度，关于它的最佳取值尚未有理论指导，大多数情况下取值为2。令 J_{FCM} 对 v_j 和 u_{ij} 求偏导，并令偏导为0，得到聚类中心和隶属度值的更新函数：

$$v_j=\frac{\sum_{i=1}^{N}u_{ij}^{m}x_i}{\sum_{i=1}^{N}u_{ij}^{m}} \tag{3}$$

$$u_{ij}=\frac{\left(\left\|x_i-v_j\right\|\right)^{-2/(m-1)}}{\sum_{g=1}^{C}\left(\left\|x_i-v_g\right\|\right)^{-2/(m-1)}} \tag{4}$$

1.2 核模糊C均值聚类算法

在原始空间下,数据点之间并非都是线性可分的，尤其对于流数据来说，数据的形式多样，即使应用较优的聚类算法，也难以得到较好的聚类效果。基于此，可利用核方法来解决这一问题，基于核方法的聚类算法[10~12]对数据的处理更加灵活，同时便于操作。核方法的核心思想是：首先通过某种非线性映射 ϕ,将原始数据嵌入到高维特征空间，使得原始空间下不能线性可分的点变得线性可分；然后利用通用的线性学习器在这个高维特征空间中对数据进行分析和处理。定义非线性映射 $\phi:X\rightarrow F$,将低维输入空间 X 映射到高维特征空间 F。核模糊 C 均值聚类算法(kernel fuzzy C-means,KFCM)的目标函数为

$$J_{KFCM}=2\sum_{j=1}^{C}\sum_{i=1}^{N}u_{ij}^{m}\left(1-k\left(x_i,v_j\right)\right) \tag{5}$$

$$\sum_{j=1}^{C}u_{ij}=1,u_{ij}\in[0,1] \tag{6}$$

$K(x_i,v_j)$ 为核函数，定义了特征空间中两点之间的欧氏离。令 J_{KFCM} 对 v_j 和 u_{ij} 求偏导，并令偏导为0，得到聚类中心和隶属度值的更新函数：

$$v_j=\frac{\sum_{i=1}^{N}u_{ij}^{m}k\left(x_i,v_j\right)x_i}{\sum_{i=1}^{N}u_{ij}^{m}k\left(x_i,v_j\right)} \tag{7}$$

$$u_{ij}=\frac{\left(1-k\left(x_i,v_j\right)\right)^{-1/(m-1)}}{\sum_{g=1}^{C}\left(1-k\left(x_i,v_g\right)\right)^{-1/(m-1)}} \tag{8}$$

其中：$k(x_i,v_j)$ 因选取的核函数而异。本文采用高斯径向基核函数（Gaussian radial bases kernels,GRBF）[13],形式如下:

$$k\left(x_i,v_j\right)=\exp\left(-\frac{\left\|x_i-v_j\right\|^2}{2\sigma^2}\right) \tag{9}$$

其中：σ 为函数的宽度参数，控制函数的径向作用范围。

1.3 统计杠杆分数

统计杠杆分数是用来衡量行向量与矩阵的一致性或相关性的标准，从而判断该向量与矩阵的相似性。统计杠杆值越高，则该行向量与矩阵中点的差异性越大，相关性越小。统计杠杆分数的应用较为广泛，在异常值检测领域[14]，用来判断外来数据是否为异常数据；在随机矩阵分析算法[15]领域，用来分析数据与随机矩阵的相关性；在矩阵一致性研究领域，如矩阵填充[16]，用于对矩阵缺失部分进行估计。

统计杠杆分数计算如下：

设矩阵 $A\in n\times d$，$A^{(i)}\in 1\times d$ 为矩阵 A 的第 i 行，矩阵 A 的第 i 行的统计杠杆分数 I 为

$$I=\left\|A^{(i)}\right\|_2^2,i\in\{1,2,\cdots,n\} \tag{10}$$

1.4 衰退聚类机制

由于流数据的动态性，随着新数据点的到达，数据模型也会发生变化。本文采用衰退聚类机制来动态更新样本核矩阵中的数据。设每一个类 $j\in[1,C]$,都被赋予一个变量值 t_j，代表被划入到第 j 类中的最后一个点的时刻，t 为新的数据点 x_t 到达的时间，在每一个数据点即将被划分到第 j 类时，本文算法采用一个单调函数 $f_j(t)$ 来计算该类的近因值[17]（近因值的概念来源于心理学的近因效应[18]，是指当人们识记一系列事物时对末

尾部分项目的记忆效果优于中间部分项目的现象），近因值越大，越能代表最新点的数据特征，受到之前数据点特征影响的程度越小。近因函数表示如下：

$$f_j(t)=\exp\left(-\gamma\left(t-t_j\right)\right) \tag{11}$$

$$\eta=\exp(-\gamma\tau),\tau\in\{1,2,3,4,5\} \tag{12}$$

其中：γ 代表一个类的衰退率[18]；参数 $\tau\in\{1,2,3,4,5\}$。本文算法将近因值小于一定阈值的类实时删除，同时用新到达的点代替该类。

2 基于差异性采样的流数据聚类算法

本文算法基于流数据的特性，采用核模糊 C 均值聚类算法为基础算法，利用差异性采样法对数据进行采样，随着流数据的流入，利用衰退聚类机制实时更新数据模型。该算法主要分为采样、聚类和更新三个阶段。

2.1 差异性采样

本文算法在对流数据进行聚类时，由于流数据的无限性，首先利用统计杠杆分数对流数据进行采样，由于聚类的类别是在核矩阵中产生，所以应使核矩阵中的采样数据之间的差异较大，从而囊括更多的数据信息，才更能代表流数据中数据的的分布情况。统计杠杆分数越高，表明数据点与原核矩阵中数据点的平均水平相差越大，同时数据在核矩阵中的影响程度越高，因此需要采集统计杠杆分数值较高的数据以保证样本核矩阵中数据点的差异性。

设样本集 S，样本中数据点的个数为 $s\left(r\le s\le R\right)$，$r$ 和 R 为自定义参数，K_{t-1} 代表 $(t-1)$ 时刻的核矩阵，$K_1=1$（将（X1，X1）带入高斯核函数公式可得）。当数据点 x_t 在 t 时刻到达时，先将其划入到样本集中得到 K_t，对其进行奇异值分解[19-20]，采用奇异值分解的原因是，在采集样本点时，需要计算核矩阵的每个行向量的统计杠杆值，计算量大，奇异值分解是提取矩阵特征的重要手段，特征值越大，说明矩阵在对应的特征向量上的方差越大，功率越大，包含的信息量也越多，同时也越能够代表数据的分布情况。因此利用奇异值分解得到核矩阵中具有代表性的 C 个向量，每次只需计算这 C 个向量和新到达点的统计杠杆值即可。以下为分解过程：

$$K_t=V_C\Sigma_C V_C^T \tag{13}$$

$$\Sigma_C=diag(\lambda_1,\cdots,\lambda_2) \tag{14}$$

$$V_C=(v_1,\cdots,v_C) \tag{15}$$

C 为聚类个数，Σ_C 是由 K_t 的最大的 C 个特征值组成的对角矩阵，$(V_C)_{s\times C}$ 是由对应的 C 个特征向量组成的矩阵。利用 V_C 来计算统计杠杆分数，目的是利用有价值的信息来挑选同样或者更加有用的信息，增加了筛选的准确率，提高了核矩阵的整体差异度。$V_C^{(i)}$ 为 V_C 的第 i 行，更新核矩阵（h 为自定义参数）：

$$K_t=\begin{cases}\begin{bmatrix}K_{t-1} & \varphi^T\\ \varphi & k\left(x_t,x_t\right)\end{bmatrix} & p_t<h\\ K_{t-1} & p_t\ge h\end{cases} \tag{16}$$

p_t 为将数据点 x_t 划入到 S 中的可能性，定义为 $(t-1)$ 时刻与 t 时刻矩阵 V_C 的统计杠杆分数的比值，定义如下：

$$p_t=\frac{\sum_{i=1}^{s}\left\|V_C^{(i)}\right\|_{2(t-1)}^2}{\sum_{i=1}^{s}\left\|V_C^{(i)}\right\|_{2(t)}^2} \tag{17}$$

统计杠杆值越大，说明新到达的点与原矩阵中的点的差异性越大，分母越大，p_t 越小；当 p_t 小于阈值 h 时，则将 t 时刻到达的点 x_t 划入到样本核矩阵中。其中阈值 $h\in(0,1]$，且 h 的取值越接近于 0，说明新数据点的到达使得统计杠杆分数越大，表明数据点与原核矩阵中数据的差异性越大，从而使得核矩阵中的点的分布范围变大，数据包含的信息越丰富，可以更有效的对流数据中的点进行聚类；但 h 的值越接近于 0，会使得满足条件的数据点变少从而需要不断筛选导致较大的计算复杂度，而且所选实验数据集的大小也会影响 h 的选取，本文算法在不同数据集上选取了不同的 h 值，详见第 3 章。本文算法的采样方法,能够减小时间复杂度和对存储空间的占用，又可以采集到相对能代表所有流数据分布的点。

2.2 聚类

利用核模糊 C 均值聚类算法将核矩阵 K_t 中的点聚成 C 个类，本文算法目标函数 J_M 如下：

$$J_M=2\sum_{j=1}^{C}\sum_{i=1}^{s}u_{ij}^m\left(1-k\left(x_i,v_j\right)\right) \tag{18}$$

令 J_M 对 v_j 和 u_{ij} 求偏导，并令偏导为 0，得到聚类中心和隶属度值的更新函数：

$$v_j=\frac{\sum_{i=1}^{s}u_{ij}^m k\left(x_i,v_j\right)x_i}{\sum_{i=1}^{s}u_{ij}^m k\left(x_i,v_j\right)} \tag{19}$$

$$u_{ij}=\frac{\left(1-k\left(x_i,v_j\right)\right)^{-1/(m-1)}}{\sum_{g=1}^{C}\left(1-k\left(x_i,v_g\right)\right)^{-1/(m-1)}} \tag{20}$$

具体聚类步骤如下：

a)给定聚类类别数 C，设定迭代收敛阈值，初始化各个聚类中心以及隶属度矩阵。

b)用当前的聚类中心根据式（20）更新隶属度矩阵，用当前隶属度矩阵根据式（19）更新各个聚类中心。

c)循环 b)，直到各个类的聚类中心不再发生变化或者隶属度矩阵的变化值小于一定的阈值，终止迭代。

2.3 动态更新数据模型

在核空间下将样本集 S 中的点聚成 C 个类后，将 C 个类映射到原空间，用来对数据流中的点进行划分，将各个点划分到距离它最近的类中。根据式（21）计算距离 t 时刻到达的点 x_t 最近的类 $j(j\in[1,C])$ 类。

$$j=\underset{j\in[1,C]}{\arg\min}\left\|v_j-x_t\right\|^2 \tag{21}$$

当得到类 j 时，计算第 j 类的近因值。若近因值小于等于于自定义阈值 η（代表一个类持续活跃的生命周期），则将该类以及类中的点从 S 中删除，并用 x_t 代替该类并作为该类的初始点；若近因值大于 η，则将 x_t 划分到该类中。因此将 x_t 划分到类中的条件为

$$\begin{cases} j = \arg\min_{j\in[1,C]} \left\| v_j - x_t \right\|^2 \\ f_j(t) > \eta \end{cases} \tag{22}$$

2.4 本文算法过程概述

本文算法的具体步骤描述如下，算法过程图示和算法流程如图 1 和 2 所示。

a)输入数据集 $X=\{x_1,x_2,\cdots,x_i,\cdots,x_N\}$，聚类个数：C，核函数：$k(x_i,v_j)$，样本集中点的个数的最大最小值：R、r,参数：τ、h。

b)初始化 U，$K_1=1$，$V_C=1$，$\Sigma_C=k(x_1,x_1)$，初始聚类中心。

c)采样:利用差异性采样方法进行采样，并利用样本集中的数据构造核矩阵。

d)聚类：采用核模糊 C 均值聚类算法对核矩阵中的数据进行聚类，得到一个带有类别标记的核矩阵。

e)划分以及动态更新数据模型：利用步骤 4 得到的标记核矩阵对流数据进行划分，同时不断更新隶属度矩阵 U。

f)输出隶属度矩阵，聚类结果。

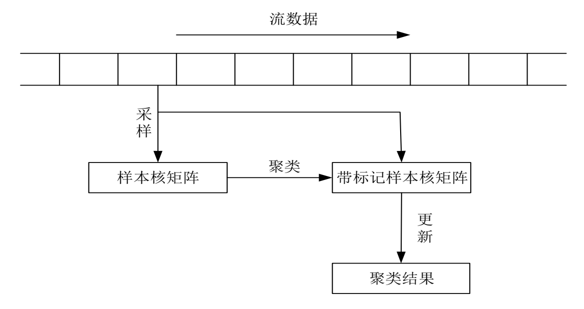


图 1 算法过程图示

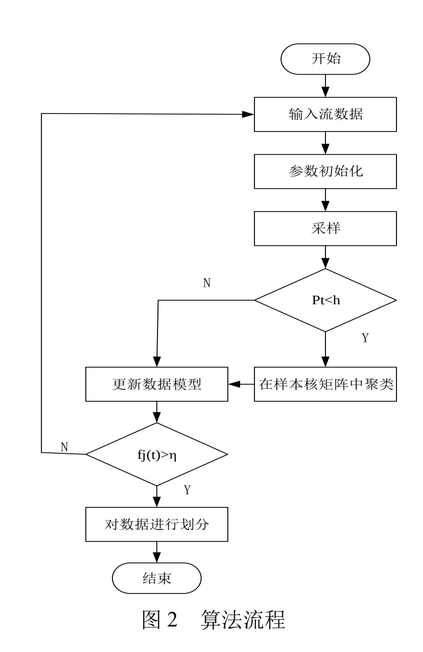


图 2 算法流程

2.5 实时动态处理概述

a)按条读取数据。为了验证本文算法的准确率，本文采用已经分类的数据集进行实验，将数据集以矩阵的形式存入，按行读取数据。

b)对数据进行采样分析，利用 MATLAB 中的功能函数 tic 函数计算每条数据的处理时间 t，并以 $t+t_0$ 作为下一条数据的初始处理时间。假设第一条数据的处理时间为 t_1，第二条数据的处理时间为 t_2，第三条数据的处理时间为 t_3 ……，则第二条数据的初始处理时间为 t_1+t_0，第三条数据的初始处理时间为 $t_2+t_1+t_0$ ……以此类推。（由于不同数据集的种类和属性的差异，因此本文算法的动态更新过程没有固定的更新频率和时间窗口）。

c)对样本核矩阵进行动态更新，每当一条数据 x_t 在 t 时刻被划入到相应的类 j 中，都要利用 t 和之前最后一个划入到该类中的数据到达的时刻 t_j 进行近因值计算，实时删除不再具有代表性的类。

2.6 时间复杂度分析

利用传统核模糊 C 均值聚类算法对流数据进行聚类，在构造核矩阵时，矩阵规模为 $N\times N$，因此算法的时间复杂度为 $O(N^2)$.在本文算法中，样本核矩阵中的数据量为 s，且 $s \ll N$，所以本文算法的时间复杂度为 $O(Ns)$，本文算法的时间复杂度远远低于传统核模糊 C 均值聚类算法。同时，在进行采样分析时，需要计算核矩阵中每行的统计杠杆分数，时间复杂度为 $O(s)$，在采用奇异值分解后，只需利用 C 个特征向量的值进行计算，且 $C<s$，时间复杂度为 $O(C)$，进一步降低了本文算法的时间复杂度。

3 实验分析

本实验的实现平台为 MATLAB2014a。为了验证本文算法的聚类效果，与 AKFCM/KFCM 和 FCM 算法分别进行实验对比，通过 AKFCM 算法对流数据聚类时采用的随机采样法与本文差异性采样法进行对比，验证本文算法的聚类效果；通过与非采样的 KFCM 算法对比，验证本文算法的时间复杂度以及聚类效果；通过与 FCM 算法进行对比，验证本文算法的聚类效果优于传统聚类算法。由于 KFCM 算法是用数据集中所有数据构造核矩阵，因此选取的数据集不宜过大，避免存储空间不足。本文选用 Movement-Libras 整个数据集，MFCC 数据集中 20 类中的部分数据，CIFAR-10 数据集中 20 类中的部分数据，Forest Cover Type 数据集中 7 类中的部分数据来模拟流数据。表 1 为实验数据集。四个数据集的长度依次变大，目的是为了验证随着流数据规模的增加，本文算法的聚类效果不会受到影响，证明本文算法对于数据量大的流数据更具有优势。本文采用归一化互信息（NMI）[21]、运行时间、准确率（A)[22]及误差平方和（SSE）作为聚类效果的评价标准。为减少偶然误差，每次实验进行 50 次取平均值。

表 1　实验数据集

数据集	属性数	类别数	数据个数
Movement-Libras	90	15	360
MFCC	22	20	3121
CIFAR-10	3072	20	10000
Forest Cover Type	4	7	20000

3.1　聚类性能指标分析

归一化互信息（NMI）在聚类中，常被用来度量某聚类算法的聚类结果与数据实际分类的相近程度，其值的范围为[0,1]，NMI 值越高，说明该聚类算法与数据的实际分类越相近，效果越好，反之效果越差。计算公式为

$$NMI=\frac{H(A)+H(B)}{H(A,B)} \tag{23}$$

$$H(A)=-\sum_{a}P_A(a)\log P_A(a) \tag{24}$$

$$H(B)=-\sum_{b}P_B(b)\log P_B(b) \tag{25}$$

$$H(A,B)=-\sum_{a,b}P_{AB}(a,b)\log P_{AB}(a,b) \tag{26}$$

其中：$H(A)$ 表示 A 向量的信息熵[23]；$H(B)$ 表示 B 向量的信息熵；$H(A,B)$ 表示 A 和 B 的联合信息熵[23]；a 、b 分别表示 A 和 B 的概率；$P_A(a)$ 、$P_B(b)$ 分别表示 A 和 B 的概率分布[24]；$P_{A,B}(a,b)$ 表示 A 和 B 的联合概率分布[24]。

准确率（A）是评价聚类结果性能最常用的准则，其计算公式如下：

$$A=\frac{\sum_{j=1}^{C}a_j}{N} \tag{27}$$

其中：N 表示实验样本数据总数；C 表示类的数目；a_j 表示聚类结果中的第 j 个聚类和实际聚类相一致的样本个数，a_j 越大，表示正确分类的样本数越多；A 越大，则聚类结果的准率越高，聚类质量越好。

误差平方和（SSE）是用来评价类间差异性的函数，公式如下：

$$SSE=\sum_{j=1}^{C}\sum_{i=1}^{d}\left\|x_{ij}-m_j\right\|^2 \tag{28}$$

其中：C 表示类的数目；d 表示第 j 类中样本的个数；x_{ij} 表示第 j 类中的第 i 个数据；m_j 表示第 j 类的聚类中心；SSE 的值越小，说明数据都被聚到相对较近的类中，聚类效果越好

3.2　参数取值分析

1）参数 τ 的取值分析

由 $\eta=\exp(-\gamma\tau)$，$\tau\in\{1,2,3,4,5\}$ 可知，τ 越大，η 越小，反之 η 越大。当近因值大于 η 时，将数据点划入到相应的类中，为了严格筛选核矩阵中的数据点，应尽量使 η 的值较大，但同时又会增加算法运行的时间，因此需要权衡时间复杂度与聚类效果之间的关系。表 2~4 分别为 τ 的取值对运行时间、NMI 以及 A 的影响。Movement-Libras 数据集的采样样本大小为 100，MFCC 数据集的采样样本大小为 500，CIFAR-10 数据集的采样样本大小为 2 000，Forest Cover Type 数据集的采样样本大小为 4 000。由表 2 可知，对于每个数据集，随着 τ 的增大，运行时间呈现不断增大的趋势，因此从运行时间上 τ 的值设为 1 较好；由表 3 可知，当 τ=1 时，四个数据集的 NMI 值都为最大，且随着 τ 值的增大，每个数据集 NMI 值降低较快；由表 4 可以看出，当 τ=1 时，四个数据集的 A 值都是最大的。因此实验中将 τ 值设为 1。

表 2　τ 的不同取值下的运行时间/ms

数据集	τ				
	1	2	3	4	5
Movement_Libras	338.21	355.48	346.09	375.42	390.72
MFCC	652.73	684.63	697.25	698.70	757.26
CIFAR-10	8975.6	8973.2	9301.4	9432.9	9804.3
Forest Cover Type	2981.7	3025.6	2911.0	3214.8	3410.5

表 3　τ 的不同取值下的 NMI 值

数据集	τ				
	1	2	3	4	5
Movement_Libras	0.8624	0.8432	0.8458	0.8063	0.8146
MFCC	0.8943	0.8656	0.8124	0.7596	0.7758
CIFAR-10	0.8793	0.8562	0.8746	0.8043	0.7859
Forest Cover Type	0.9025	0.8925	0.8856	0.8293	0.7894

表 4　τ 的不同取值下的 A 值

数据集	τ				
	1	2	3	4	5
Movement_Libras	0.9073	0.8825	0.8401	0.8190	0.7911
MFCC	0.9174	0.8941	0.8631	0.8854	0.8126
CIFAR-10	0.8992	0.8829	0.8944	0.8788	0.8635
Forest Cover Type	0.9130	0.8799	0.8991	0.8432	0.8673

2）参数 h 的取值分析

参数 $h\in(0,1]$，h 的取值越小，说明新数据点的到达使得统计杠杆分数越大，表明数据点与核矩阵中原数据点的差异性越大，从而使得核矩阵中的点的分布范围变大，数据包含的信息越丰富，但 h 的值越小，会使得满足条件的数据点变少从而需要不断筛选造成较大的时间复杂度，因此需要根据不同的数据集，通过实验验证，权衡时间复杂度与聚类效果之间的关系来确定 h 的值。由表 5，NMI 和 A 的值随着 h 的增大而变小，运行时间也变小，且变化值较小，又由于 Movement_Libras 数据集规模较小，运行时间较短，因此采用聚类结果最准确的值 $h=0.1$。由表 6，NMI 和 A 的值在 $h=0.4$ 时降低幅度较大，且在 $h=0.1$、0.2、0.3 时的值相差较小；运行时间上，当 $h=0.3$ 时，

运行时间明显降低，并且之后趋于平缓，因此在 MFCC 数据集上，h 值设为 0.3。由表 7，当 NMI 最大时 $h=0.4$，且之后 NMI 值逐渐降低，且降幅较大；运行时间上呈现逐渐减小的趋势，且当 $h=0.1、0.2、0.3$ 时，运行时间很大，$h=0.4$ 时降幅较大，且之后趋于平缓，同时 $h=0.4$ 时的 A 值与最大值相差不多，因此在 CIFAR-10 数据集上，将 h 设为 0.4。由表 8，运行时间在 $h=0.2$ 时降低幅度较大且之后趋于平缓；同时 NMI 和 A 的值逐渐降低，且在 $h=0.2$ 时，NMI 和 A 值相对较大，因此在 Forest Cover Type 数据集上，将 h 设为 0.2。

表 5　Movement_Libras 数据集 h 取值分析

h	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
NMI	0.8624	0.8502	0.8594	0.8006	0.8399	0.8458	0.8303	0.8242	0.8112
A	0.9073	0.8992	0.8801	0.8688	0.8597	0.8525	0.8457	0.8391	0.8361
运行时间/ms	338.21	336.12	331.25	321.92	318.60	309.19	311.32	299.30	290.56

表 6　MFCC 数据集 h 取值分析

h	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
NMI	0.9032	0.8982	0.8943	0.7206	0.7299	0.6958	0.6803	0.6542	0.6212
A	0.9198	0.9183	0.9174	0.8233	0.7980	0.7815	0.7570	0.7499	0.7161
运行时间/ms	1176.9	976.61	652.73	654.90	643.52	662.15	619.04	629.18	600.79

表 7　CIFAR-10 数据集 h 取值分析

h	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
NMI	0.8724	0.8602	0.8794	0.8793	0.7499	0.7158	0.6803	0.6842	0.6512
A	0.8669	0.8640	0.8571	0.8992	0.7350	0.7375	0.7216	0.6991	0.6761
运行时间/ms	14562	12729	12382	8975.6	8709.4	8577.0	8548.1	8762.3	8434.6

表 8　Forest Cover Type 数据集 h 取值分析

h	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
NMI	0.9094	0.9025	0.9000	0.8606	0.8432	0.8400	0.8207	0.8233	0.8142
A	0.9109	0.9130	0.8873	0.8988	0.8596	0.8432	0.8592	0.8348	0.8293
运行时间/ms	4019.2	2981.7	2810.5	2699.0	2579.1	2600.7	2583.6	2401.9	2481.6

3.3　聚类性能分析

1）　NMI 值分析

图 3~6 分别为四种算法在不同数据集上的 NMI 值。由于本文算法和 AKFCM 算法需要进行采样，样本大小不同，聚类效果也不同，所以此 NMI 值是变化的，而 KFCM 和 FCM 算法为非采样算法，因此为固定值。通过分析图 3~6 可知：a）本文算法的值始终大于 AKFCM 算法的值，且随着数据规模的不断扩大，两者差值逐渐增大，证明本文算法采用的采样方法优于 AKFCM 算法采用的随机采样法；b）在四个数据集上，本文算法的 NMI 值都高于 KFCM 和 FCM 算法的值，且远远高于 FCM 算法的值，同时随着采样数目的增加，本文算法 NMI 值逐渐增大，证明在对流数据进行聚类上，本文算法优于传统聚类算法。

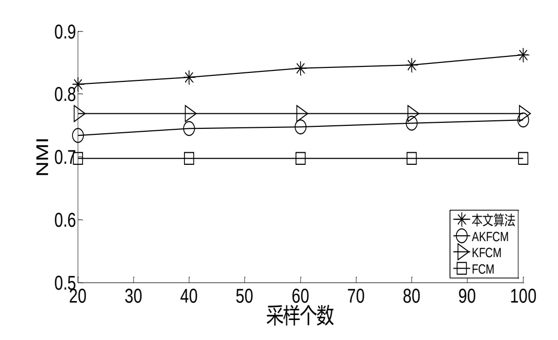


图 3　Movement_Libras 数据集对比实验 NMI 值

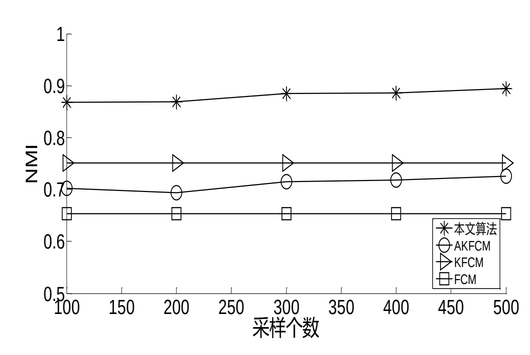


图 4　MFCC 数据集对比实验 NMI 值

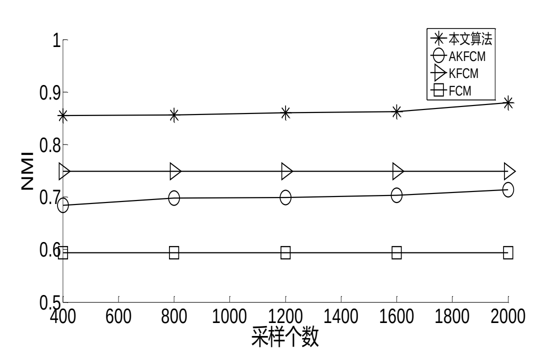


图 5　CIFAR-10 数据集对比实验 NMI 值

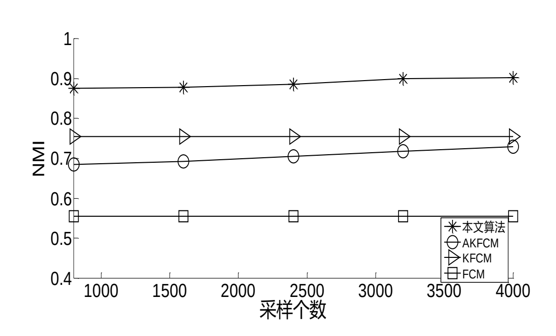


图 6　Forest Cover Type 数据集对比实验 NMI 值

2）其他性能比较分析

下面主要从运行时间（time/ms）、误差平方和（SSE）和准确率（A）三个方面进行对比分析。四组数据集样本数量分别为 100、500、2 000、4 000。从表 9~12 可以看出，在运行时间（time/ms）、误差平方和（SSE）以及准确率（A）三个方面本文算法都要优于 AKFCM 和 KFCM，且随着数据集规模的扩大，

AKFCM 与 KFCM 算法的准确率逐渐降低，本文提出的算法仍然具备较高的准确率。证明本文算法的聚类效果优于利用随机采样法对流数据进行聚类的算法。虽然在运行时间上本文算法要高于 FCM 算法，但在误差平方和以及准确率上本文算法要远优于 FCM 算法。

表 9　Movement_Libras 数据集运行时间、SSE 和 A

	核矩阵样本数	Time/ms	SSE	A
本文算法	100	338.21	12985	0.9073
AKFCM	100	349.00	14260	0.7659
KFCM	全部	565.75	14098	0.8509
FCM	0	187.17	15780	0.6134

表 10　MFCC 数据集运行时间、SSE 和 A

	核矩阵样本数	Time/ms	SSE	A
本文算法	500	652.73	151750	0.9174
AKFCM	500	786.79	186400	0.7284
KFCM	全部	1054.3	174208	0.8273
FCM	0	287.9	245780	0.5934

表 11　CIFAR-10 数据集运行时间、SSE 和 A

	核矩阵样本数	Time/ms	SSE	A
本文算法	2000	8975.6	484679	0.8992
AKFCM	2000	9478.1	683679	0.6734
KFCM	全部	16445	637863	0.7348
FCM	0	2480.0	977894	0.4122

表 12　Forest Cover Type 数据集运行时间、SSE 和 A

	核矩阵样本数	Time/ms	SSE	A
本文算法	4000	2981.7	64846	0.9130
AKFCM	4000	3975.6	82370	0.6734
KFCM	全部	5776.0	84579	0.7763
FCM	0	923.0	107894	0.4950

4　结束语

本文提出了一种基于差异性采样的流数据聚类算法，在采样阶段，利用统计杠杆分数衡量数据点与原样本集中点的差异性，得到一个数据点之间差异性较大的样本核矩阵，使样本中的点更能代表流数据中点的分布特征；在数据更新阶段，本文采用衰退聚类机制，随着新数据点的到达，实时删除无法反映新数据点特征的类，并用新数据点代替该类，以保证实时分析得到更能代表所有数据分布的数据模型。实验结果表明，本文算法在保证聚类效果的前提下，大大降低了对流数据聚类的时间复杂度；同时随着数据集规模的扩大，本文算法的聚类效果并未受到影响，证明本文算法对于数据量大的流数据更具有优势。

参考文献：

[1] Aggarwal C C, Han Jiawei, Wang Jianyong, *et al.* A framework for clustering evolving data streams [C]// Proc of the 29th International Conference on Very Large Data Bases. Berlin: Morgan Kaufmann Publishers, 2003: 81-92.

[2] Jain A, Zhang Z, Chang E Y, Adaptive non-linear clustering in data streams [C]// Proc of International Conference on Information and Knowledge Management. 2006: 122-131.

[3] Chen Yixin, Li Tu. Density-based clustering for real-time stream data [C]// Proc of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2007: 133-142.

[4] Havens T C, Chitta R, Jain A K, *et al.* Speedup of fuzzy and possibilistic kernel C-means for large-scale clustering [C]// Proc of IEEE International Conference on Fuzzy Systems. 2011: 463-470.

[5] 张中平，王浩，薛伟，等. 动态滑动串口的数据流聚类方法 [J]. 计算机工程与应用, 2011, 47 (7): 135-138.

[6] 黄红艳，安素芳. 数据流聚类算法在入侵检测中的应用 [J]. 计算机工程与应用, 2012, 48 (20): 112-116.

[7] Ester M, Kriegel H P, Sander J, *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise [C]// Proc of KDD. 1996: 226-231.

[8] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, *et al.* Fast approximation of matrix coherence and statistical leverage [C]// Proc of Journal of Machine Learning Research. 2012: 3475-3506.

[9] Goyal L M, Mittal Mamta, Sethi J K. Fuzzy model generation using subtractive and fuzzy C-means clustering [J]. CSI Trans on ICT, 2016, 4 (2-4) , 129-133.

[10] 范子静，罗泽，马永征. 一种基于模糊核聚类的谱聚类算法 [J]. 计算机工程, 2017, 43 (11): 161-165, 172.

[11] 张腾达，吕晓琪，任晓颖，等. 基于空间模糊核聚类的脑肿瘤图像分割方法 [J]. 控制工程, 2017, 24 (10): 2107-2111.

[12] 王书文，皮炳坤，张弘强，等. 一种基于模糊核聚类算法的图像分类方法 [J]. 西北师范大学学报：自然科学版, 2016, 52 (5): 42-45.

[13] Yin Yong, Hao Yinfeng, Bai Yu, *et al.* A Gaussian-based kernel Fisher discriminant analysis for electronic nose data and applications in spirit and vinegar classification [J]. Journal of Food Measurement and Characterization, 2017, 11 (1): 24-32.

[14] Hoaglin D C, Welsch R E. The hat matrix in regression and ANOVA [J]. The American Statistician, 1978, 32 (1): 17-22.

[15] Boutsidis C, Mahoney M W, Drineas P. An improved approximation algorithm for the column subset selection problem [C]// Proc of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms. 2009: 968-977.

[16] Recht B. Exact matrix completion via convex optimization [J]. ACM, 2012, 55 (6): 111-119.

[17] Aggarwal C C. Data streams: models and algorithms. Springer Science

[EB/OL]. (2007) . http://www.stat.wvu.edu/~jharner/courses/stat624/docs/streambook. pdf.

[18] 张军. 近因效应的认知影响及教学应用 [J]. 化学学, 2017, 36 (9): 24-28.

[19] Wang Shuli, Wang Guanxiang. Texture classification by bit-plane multifractal spectrum and bit-plane barycentric coordinates of wavelet coefficients based on SVD [EB/OL]. (2017) . http://dpi-proceedings.com/index. php/dtcse/article/download/8875/8444.

[20] 邱志伟，岳顺，岳建平，等. 基于奇异值分解 (SVD) 的桥梁监测数据去噪方法 [J]. 工程勘察, 2017, 45 (12): 36-39.

[21] Rand W M. Objective criteria for the evaluation of clustering methods [J]. Journal of the American Statistical Association, 1971, 66 (1): 846-850.

[22] Sun Y, Zhu Q M, Chen Z X. An iterative initial points refinement algorithm for categorical data clustering [J]. Pattern Recognition Letters, 2002, 23 (7): 875-884.

[23] 王天志，吴仕勇，陈恳，等. 基于联合信息熵和粗糙集理论的关联知识发现 [J]. 云南民族大学学报：自然科学版, 2010, 19 (3): 224-227.

[24] Zhang Jinping, Zhao Yong, Ding Zhihong. Research on the joint probability distribution of rainfall and reference crop evapotranspiration [J]. Paddy and Water Environment, 2017, 15 (1) , 193-200.